Project 1: Robot Path Planning

Group Members: Anna Teng, Sunny Li

Program Running Instructions:

Program should be run via the terminal with 4 arguments being the python file containing the project code, input file path, k, and output file path in the format: **python3 <robot_path_finding_code_file> <input_file_path> <k_value> -o <output_file_path>**.

For example, with input 1 and k of 0:

python3 RobotPathPlanning.py "Inputs/Input3.txt" 0 -o Output3.txt

There are also 3 folders in the same directory as the code file that we utilized when making/testing the project:

1. Inputs: stores all the inputs
2. Outputs: the folder the outputs are generated to
3. Logs: stores all logs generated during testing

Program Source Code:

```
"""
Robot Path Finding Project
Authors: Anna Teng, Sunny Li
"""

import os
import logging
import heapq
import math
import copy
import argparse
from datetime import datetime


DIRECTIONS = [(1, 0), (1, 1), (0, 1), (-1, 1), (-1, 0), (-1, -1), (0, -1),
(1, -1)]


# Set up logging configuration
def setup_logging(input_file_name, output_file_name, k):
    os.makedirs('Logs', exist_ok=True)
    timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
    log_filename = os.path.join('Logs', f"{timestamp}.log")

    logging.basicConfig(
        filename=log_filename,
        level=logging.INFO,
        format='%(message)s'
    )
```

```
    logging.info("========= A* Search Run Log =========")
    logging.info(f"Input File: {input_file_name}")
    logging.info(f"Output File: Outputs\\{output_file_name}")
    logging.info(f"k: {k}")
    logging.info("====================================\n")


# OutputModel class to simplify the code relating to the output
class OutputModel:
    def __init__(self, output_dict: dict):
        self.depth = output_dict["depth"]
        self.generated_nodes = len(output_dict["generated_nodes"])
        self.moves = " ".join(map(str, output_dict["moves"]))
        self.f_values = " ".join(map(str, output_dict["f_values"]))

        rows = [" ".join(map(str, row)) for row in output_dict["workspace"]]
        self.output_workspace = "\n".join(rows)


# Node class
class Node:
    def __init__(self, pos, path_cost, total_cost, last_angle=0,
parent=None):
        self.pos = pos
        self.path_cost = path_cost
        self.total_cost = total_cost
        self.last_angle = last_angle
        self.parent = parent

    def __lt__(self, other):
        # Used for heapq
        return self.total_cost < other.total_cost

    def __repr__(self):
        # Used for debugging
        return str(self.pos)


def process_input(input_file_path):
    """
    Used to process input file into workable data
    :param input_file_path: path of input file
    :return: tuple of starting and goal positions, and a 2D array of the
robot workspace
    """
    try:
        input_file = open(input_file_path, 'r')
    except FileNotFoundError:
        print("File not found!")
        return
    first_line = input_file.readline()
    first_line_data = first_line.strip().split()
    start_pos = (int(first_line_data[0]), int(first_line_data[1]))
    goal_pos = (int(first_line_data[2]), int(first_line_data[3]))
    workspace = []
    for line in input_file:
```

```
        if line != "\n":
            curr_line = []
            nums = line.strip().split()
            for num in nums:
                curr_line.append(int(num))
            workspace.append(curr_line)
    input_file.close()
    workspace.reverse()
    return start_pos, goal_pos, workspace


def calculate_heuristic(curr_pos, goal_pos):
    """
    Calculate the h(n) value based on current and goal position
    :param curr_pos: tuple of current location
    :param goal_pos: tuple of goal position
    :return: The heuristic value calculated
    """
    return math.sqrt((curr_pos[0]-goal_pos[0])**2+(curr_pos[1]-
goal_pos[1])**2)


def is_valid_pos(pos, workspace):
    """
    Returns whether a position is valid (not out of bound and not blocked)
    :param pos: position to check
    :param workspace: 2D list of workspace
    :return: boolean of validity
    """
    if pos[1] < 0 or pos[0] < 0 or pos[1] >= len(workspace) or pos[0] >=
len(workspace[0]):
        return False
    return workspace[pos[1]][pos[0]] != 1


def calculate_angle_cost(curr_angle, new_angle, k):
    """
    Calculates the angle cost for changing direction based on curr_angle and
new_angle
    :param curr_angle: angle of current node in degrees
    :param new_angle: angle of next node in degrees
    :param k: penalty coefficient for direction change
    :return: angle cost as a float
    """
    delta_theta = abs(new_angle - curr_angle)
    if delta_theta > 180:
        delta_theta = 360 - delta_theta

    return k * (delta_theta / 180)


def calculate_distance_cost(curr_node, new_pos):
    """
    Calculates the distance cost to get to new_pos from curr_pos
    :param curr_node: coordinates of curr position
    :param new_pos: coordinates of new position
    :return: distance cost as a float
```

```
    """
    # Check if horizontal or vertical move
    if abs(new_pos[0] - curr_node.pos[0]) + abs(new_pos[1] -
curr_node.pos[1]) == 1:
        distance_cost = 1
    else:
        distance_cost = math.sqrt(2)

    return distance_cost


def calculate_step_cost(curr_node, new_pos, k):
    """
    Calculates the step cost by adding distance and angle costs
    :param curr_node: coordinates of curr position
    :param new_pos: coordinates of new position
    :param k: penalty coefficient for direction change
    :return: total step cost as a float
    :return: new angle in degrees
    """
    dx, dy = new_pos[0] - curr_node.pos[0], new_pos[1] - curr_node.pos[1]
    new_angle = math.degrees(math.atan2(dy, dx)) % 360

    if curr_node.parent is None:
        angle_cost = 0
        new_angle = 0
    else:
        angle_cost = calculate_angle_cost(curr_node.last_angle, new_angle, k)

    distance_cost = calculate_distance_cost(curr_node, new_pos)

    return distance_cost + angle_cost, new_angle


def a_star_search_algo(start_pos, goal_pos, workspace, k):
    """
    Implementation of the A* search algorithm
    :param start_pos: starting coordinate
    :param goal_pos: goal coordinate
    :param workspace: workspace 2D list
    :return: None if no solution; curr_node, generated if solution is found
    """
    start_node = Node(start_pos, 0, calculate_heuristic(start_pos, goal_pos))
    reached = {start_pos: start_node.total_cost}
    frontier = []
    heapq.heappush(frontier, start_node)

    logging.info(f"Generated node:\t\t{start_node}")

    while frontier:
        # Get the smallest value
        curr_node = heapq.heappop(frontier)

        logging.info(f"Frontier popped:\t{curr_node}")

        # If solution is found
        if curr_node.pos == goal_pos:
```

```
            logging.info("======GOAL REACHED!!!======")
            logging.info("Reached: " + str(len(reached)))
            return curr_node, reached

        # Generate all child nodes
        for direction in DIRECTIONS:
            new_pos = (curr_node.pos[0] + direction[0], curr_node.pos[1] +
direction[1])

            # Append node to generated and frontier if it's valid
            if is_valid_pos(new_pos, workspace):
                step_cost, new_angle = calculate_step_cost(curr_node,
new_pos, k)
                child_path_cost = curr_node.path_cost + step_cost
                child_heuristic = calculate_heuristic(new_pos, goal_pos)
                child_total_cost = child_path_cost + child_heuristic
                if new_pos in reached and reached[new_pos] <=
child_total_cost:
                    continue
                child_node = Node(new_pos, child_path_cost, child_total_cost,
last_angle=new_angle, parent=curr_node)
                reached[child_node.pos] = child_node.total_cost
                heapq.heappush(frontier, child_node)

                logging.info(f"Generated node:\t\t{child_node}")
                logging.info(f"Added to frontier:\t{child_node}")


def calculate_output_values(final_node, workspace):
    """
    Function used to calculate several values needed for output
    :param final_node: The last node in the path found
    :param workspace: 2D list of the workspace
    :return: dictionary of all the values
    """
    # Just didn't want to change the original workspace
    new_workspace = copy.deepcopy(workspace)

    curr_node = final_node
    depth = -1
    moves = []
    f_values = []

    while curr_node:
        pos = curr_node.pos
        if new_workspace[pos[1]][pos[0]] != 2 and
new_workspace[pos[1]][pos[0]] != 5:
            new_workspace[pos[1]][pos[0]] = 4
        depth += 1
        f_values.append(curr_node.total_cost)

        if curr_node.parent is not None:
            direction = (curr_node.pos[0] - curr_node.parent.pos[0],
                         curr_node.pos[1] - curr_node.parent.pos[1])
            move = DIRECTIONS.index(direction)
            moves.append(move)
```

```
        curr_node = curr_node.parent

    moves.reverse()
    f_values.reverse()
    new_workspace.reverse()

    return {
        "depth": depth,
        "moves": moves,
        "f_values": f_values,
        "workspace": new_workspace
    }


def output_into_file(output: OutputModel, file_name):
    """
    Used to write all output data into a output file
    :param output: The OutputModel used to help with output generation
    :param file: the filepath of the output file
    :return: None
    """
    os.makedirs('Outputs', exist_ok=True)

    # Construct the full path
    output_file_path = os.path.join('Outputs', file_name)

    output_file = open(output_file_path, "w")

    print(output.depth, file=output_file)
    print(output.generated_nodes, file=output_file)
    print(output.moves, file=output_file)
    print(output.f_values, file=output_file)
    print(output.output_workspace, file=output_file)

    output_file.close()


def main():
    parser = argparse.ArgumentParser(description="Run A* search on a robot
workspace")
    parser.add_argument("input_file", type=str, help="Path to the input
file")
    parser.add_argument("k", type=int, nargs='?', default=0, help="Angle cost
penalty parameter (default: 0)")
    parser.add_argument("-o", "--output_file", type=str,
default="Output.txt", help="Name of the output file (default: 'Output.txt')")
    args = parser.parse_args()

    setup_logging(args.input_file, args.output_file, args.k)

    start_pos, goal_pos, workspace = process_input(args.input_file)
    result = a_star_search_algo(start_pos, goal_pos, workspace, args.k)
    if result:
        final_node, generated_nodes = result
        output_dict = calculate_output_values(final_node, workspace)
        output_dict["generated_nodes"] = generated_nodes
        output = OutputModel(output_dict)
```

```
        output_into_file(output, args.output_file)


if __name__ == "__main__":
    main()
```

Program Output Files:

Input 1 K = 0:

31
194
0 0 7 7 7 7 7 7 7 7 7 0 0 0 0 1 0 0 0 0 0 0 0 7 7 0 0 0 0 0 7 0
32.57299494980466 32.622776601683796 32.67572330035593 32.82509590207858
32.988682805403634 33.1684647227918 33.366774513857266 33.586369156128
33.830516434096076 34.103098247786185 34.40873160871375 34.413459741226546
34.41868167352756 34.4244787752596 34.43095126760845 35.006742657457565
35.02498060213621 35.045743124634214 35.06958612548419 35.097238938210836
35.12967631234924 35.16822857026841 35.214755041863 35.388346874966274
35.627416997969526 35.63911171640227 35.65536869969684 35.67945481172171
35.71862684627243 35.792417163603844 35.97056274847714 35.97056274847714
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 2 4 4 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 4 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 4 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 4 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 4 0 1 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 4 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 4 1 1 0 0 0 4 4 4 4 4 4 4 4 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 0 4 4 4 4 4 1 0 0 0 0 0 0 0 4 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 4 4 4 4 4 4 1 0 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 4 5 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Input 1 K = 2:

31
341
7 0 0 7 7 7 7 7 7 7 0 0 0 0 1 0 0 0 0 0 0 0 7 7 0 0 0 0 0 7 0
32.57299494980466 32.73513308910474 32.778666463751044 32.82509590207858
33.488682805403634 33.6684647227918 33.866774513857266 34.086369156128
34.330516434096076 34.603098247786185 34.90873160871375 35.413459741226546
35.41868167352756 35.4244787752596 35.43095126760845 36.506742657457565
37.02498060213621 37.045743124634214 37.06958612548419 37.097238938210836
37.12967631234924 37.16822857026841 37.214755041863 37.888346874966274
38.127416997969526 38.63911171640227 38.65536869969684 38.67945481172171
38.71862684627243 38.792417163603844 39.47056274847714 39.97056274847714
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 2 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 4 4 4 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 4 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 4 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 4 0 1 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 4 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 4 1 1 0 0 0 4 4 4 4 4 4 4 4 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 0 4 4 4 4 4 1 0 0 0 0 0 0 4 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 1 1 0 0 0 1 1 1 0 0 1 1 0 0 4 4 4 4 4 4 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0

0 0 0 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 4 5 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Input 1 K = 4:

31
366
7 0 0 7 7 7 7 7 7 7 7 7 0 7 7 7 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
32.57299494980466 32.73513308910474 32.778666463751044 32.82509590207858
33.988682805403634 34.1684647227918 34.366774513857266 34.586369156128
34.830516434096076 35.103098247786185 35.40873160871375 35.75290645585865
36.14213562373095 37.14213562373095 38.58573555203046 39.09507824507425
39.681834851628594 40.7025973741266 40.726440374976576 40.75409318770322
40.78653056184162 40.825082819760794 40.871609291355384 40.928780056167774
41.00054941671415 41.092980243349615 41.21572820569554 41.38477631085024
41.627416997969526 42.627416997969526 42.62741699796952 42.62741699796952
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 2 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 4 4 4 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 4 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 4 0 0 1 1 0 0 1 1 0 1 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 4 0 1 1 1 0 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 4 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 4 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 0 4 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 1 1 4 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 4 4 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 5 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 1 1 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 4 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 4 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 1 0 0 1 0 4 0 0 0 0 0 0 0 0 1 1 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 4 4 4 4 4 4 4 4 4 4 4 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Input 2 K = 0:

37
380
0 0 7 0 0 1 1 1 1 1 1 0 0 0 1 1 1 2 2 2 1 0 0 0 1 1 1 2 2 2 1 1 2 1 0 1 0
37.8021163428716 38.013511046643494 38.235341863986875 39.538997298749976
39.797825588281356 40.06966046470001 40.069967401935514 40.07030161279838
40.07066690098348 40.071067811865476 40.071509822506016 40.07199959321647
40.35533905932738 40.65833174289156 40.98268409419626 40.98527659649403
40.98821368682716 40.991568865010166 41.24710879827376 41.52691193458119
41.83394163668508 41.83516978220376 42.112698372208094 42.42241793342256
42.76901958965595 42.77681122334285 42.78653056184162 42.798989873223334
43.01853433051622 43.2842712474619 43.60923954912999 43.616327673029275
43.62741699796952 44.0995529529691 44.20390822051099 44.2776985378424
44.4558441227157 44.4558441227157
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 0 0 4 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 4 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 4 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 4 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 4 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 4 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 4 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 4 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 4 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 4 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 4 4 4 4 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 1 0 0 1 1 4 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 1 4 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 1 0 4 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 4 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 4 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 1 1 4 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 4 4 4 4 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 1 1 0 1 1 1 0 1 1 0 4 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 4 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 4 0 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 4 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 2 4 4 1 1 1 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 4 4 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Input 2 K = 2:

37
496
7 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 1 2 2 2 1 0 0 0 1 1 1 2 2 2 2 1 1 1 1 0 0
37.8021163428716 39.05727401181051 39.29239139154464 39.538997298749976
39.797825588281356 40.06966046470001 40.35533905932737 40.85533905932738
40.85533905932738 40.85533905932738 40.85533905932738 40.85533905932738
40.85533905932738 41.65833174289156 41.98268409419626 42.48527659649403
42.48821368682716 42.491568865010166 43.24710879827376 43.52691193458119
43.83394163668508 44.33516978220376 45.112698372208094 45.42241793342256
45.76901958965595 46.27681122334284 46.286530561841616 46.29898987322333
47.01853433051622 47.284271247461895 47.60923954912998 48.0100924241513
48.54415533044172 48.59955295296909 48.70390822051098 48.955844122715696
49.455844122715696 49.455844122715696
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 0 4 4 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 4 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 4 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 4 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 4 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 4 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 4 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 4 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 1 4 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 4 4 4 4 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 1 0 0 1 1 4 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 1 4 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 1 0 4 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 4 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 4 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 1 1 4 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 4 4 4 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 4 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 0 4 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 4 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 4 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 2 0 0 1 1 1 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 4 4 4 4 4 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Input 2 K = 4:

37
602
7 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 1 2 2 2 1 0 0 0 1 1 1 2 2 2 2 1 1 1 1 0 0
37.8021163428716 39.05727401181051 39.29239139154464 39.538997298749976
39.797825588281356 40.06966046470001 40.35533905932737 41.35533905932738
41.35533905932738 41.35533905932738 41.35533905932738 41.35533905932738
41.35533905932738 42.65833174289156 42.98268409419626 43.98527659649403
43.98821368682716 43.991568865010166 45.24710879827376 45.52691193458119
45.83394163668508 46.83516978220376 48.112698372208094 48.42241793342256
48.76901958965595 49.77681122334284 49.786530561841616 49.79898987322333
51.01853433051622 51.284271247461895 51.60923954912998 52.0100924241513
53.04415533044172 53.09955295296909 53.20390822051098 53.455844122715696
54.455844122715696 54.455844122715696
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 0 4 4 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 4 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 4 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 4 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 4 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 4 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 4 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 4 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 4 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 4 4 4 4 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 1 0 0 1 1 4 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 1 4 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 1 0 4 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 4 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 4 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 1 1 4 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 4 4 4 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 4 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 0 4 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 4 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 4 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 2 0 0 1 1 1 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 4 4 4 4 4 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Input 3 K = 0:

48
394
1 1 2 2 2 2 2 1 0 0 1 1 1 1 2 1 0 0 0 0 7 0 0 0 0 0 7 0 0 0 1 1 1 1 0 0 0 0 0 0 1 0 1 0 1 0 0 1
46.51881339845203 46.602707673153105 46.69185152366881 47.29105474894765
47.90974558182222 48.548445851333845 49.207667325580374 49.88790881437237
50.047052213325614 50.092358377461835 50.13994136467273 50.299914698929804
50.47161725826807 50.656292164376524 50.85534862931096 51.633750523477666
51.87134969118418 51.88901530667865 51.90782558054147 51.92789428890646
51.94935062553747 52.620573423320565 52.65374609703038 52.689432032852196
52.72792206135786 52.76955262170047 52.81471482258825 53.614340677975186
53.68279485226897 53.75766375181925 53.83985122732316 53.94112549695428
54.0585702391122 54.196164611692126 54.359209651784354 54.400460381958766
54.44810051389684 54.503689783942946 54.56931948749233 54.64784767501942
54.74326178322248 54.82897654146032 54.921407368095785 55.012581805104006
55.099552952969084 55.203908220510975 55.27769853784238 55.45584412271569
55.45584412271569
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 4 4 4 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 4 4 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 1 0 4 4 0 0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 4 4 4 4 4 4 4 0 0 0 0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 4 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 4 4 4 4 4 1 1 0 0 1 1 1 0 0 1 0 4 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 4 1 1 0 0 0 4 4 4 4 4 4 1 0 1 1 4 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 4 0 1 0 0 0 0 0 0 0 0 0 4 4 4 4 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 4 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 4 0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0

```
0 0 0 0 0 0 0 1 0 4 0 0 0 0 1 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 4 4 4 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 4 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 4 1 1 1 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 1 1 4 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 4 1 1 1 0 1 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 4 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 4 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 4 0 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 2 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Input 3 K = 2:

```
48
506
1 1 2 2 2 2 2 1 0 0 1 1 1 1 2 1 0 0 0 0 7 7 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1
46.51881339845203 46.602707673153105 47.19185152366881 48.29105474894765
48.90974558182222 49.548445851333845 50.207667325580374 50.88790881437237
51.547052213325614 52.092358377461835 52.13994136467273 52.799914698929804
52.97161725826807 53.156292164376524 53.35534862931096 54.633750523477666
55.37134969118418 55.88901530667865 55.90782558054147 55.92789428890646
55.94935062553747 57.120573423320565 57.84507664120184 58.39094512054433
58.44035690507799 58.49372694750279 58.5515354451702 58.61434067797518
58.68279485226897 58.75766375181925 58.83985122732316 59.44112549695428
59.5585702391122 59.69616461692126 59.859209651784354 60.400460381958766
60.44810051389684 60.503689783942946 60.56931948749233 60.64784767501942
60.74326178322248 60.86124762152187 61.010092424151296 61.20211411065617
61.4558441227157 61.955844122715696 61.955844122715696 61.95584412271569
61.95584412271569
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 4 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 4 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 4 4 4 4 4 4 4 4 4 4 4 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 4 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 4 4 4 4 4 1 1 0 0 1 1 1 0 0 1 0 4 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0 0 0 1 1 4 1 1 0 0 0 4 0 0 0 0 0 1 0 1 1 4 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 4 0 1 0 0 0 0 4 4 4 4 4 4 4 4 4 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 4 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 4 0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 4 0 0 0 0 1 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 4 4 4 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 4 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 4 1 1 1 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 1 1 4 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 4 1 1 1 0 1 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 4 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 4 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 4 0 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 2 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Input 3 K = 4:

```
48
653
1 1 2 2 2 2 2 1 0 0 1 1 1 1 2 1 0 0 0 0 7 7 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1
46.51881339845203 46.602707673153105 47.69185152366881 49.29105474894765
49.90974558182222 50.548445851333845 51.207667325580374 51.88790881437237
53.047052213325614 54.092358377461835 54.13994136467273 55.2999146989298
55.47161725826807 55.656292164376524 55.85534862931096 57.633750523477666
58.87134969118418 59.88901530667865 59.90782558054147 59.92789428890646
59.94935062553747 61.620573423320565 62.34507664120184 63.39094512054433
63.44035690507799 63.49372694750279 63.5515354451702 63.61434067797518
63.68279485226897 63.75766375181925 63.83985122732316 64.94112549695427
65.0585702391122 65.19616461169213 65.35920965178435 66.40046038195877
66.44810051389683 66.50368978394295 66.56931948749232 66.64784767501942
66.74326178322248 66.86124762152187 67.0100924241513 67.20211411065617
67.4558441227157 68.4558441227157 68.45584412271569 68.45584412271569
68.45584412271569
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 4 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 4 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 4 4 4 4 4 4 4 4 4 4 4 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 4 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 4 4 4 4 4 1 1 0 0 1 1 1 0 0 1 0 4 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 4 1 1 0 0 0 4 0 0 0 0 0 1 0 1 1 4 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 4 0 1 0 0 0 0 4 4 4 4 4 4 4 4 4 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 4 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 4 0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 4 0 0 0 0 1 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 4 4 4 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 4 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 4 1 1 1 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 1 1 4 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 4 1 1 1 0 1 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 4 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 4 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 4 0 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 2 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```